Spring 5-1-2006

USDSU Online/Interactive Student Bulletin Board

Brenda VanOrmer
Dakota State University

USDSU Online/Interactive Student Bulletin Board

A graduate project submitted to Dakota State University in partial fulfillment of the requirements for the degree of

Master of Science

in

Information Systems

May, 2006

By
Brenda VanOrmer

Project Committee:

Dr. Wayne Pauli
Tom Farrell
Jason Shea

We certify that we have read this project and that, in our opinion, it is satisfactory in scope and quality as a project for the degree of Master of Science in Information Systems.

Project Committee

Faculty supervisor: Wayne E Tarrell Date: 5/3/06

Committee member: Jason Shea Date: 5/3/06

Committee member: Tom Farrell Date: 5/3/06

ACKNOWLEDGMENT

I would like first like to thank Dr. Omar El-Gayar for his continued support and encouragement which without I may not have made it this far. He has a vested interest in all of his students and I appreciate and value his support.

I would also like to thank my committee members, Dr. Wayne Pauli, Tom Farrell, and Jason Shea for agreeing to work on my committee. I realize that their schedules are sometimes hectic and I appreciate them taking on this extra duty.

Additional appreciation goes out to the USDSU staff for allowing me to complete this project in fulfillment of my master's project requirement. Thank you to Deanna Kost at USDSU for initiating this project and allowing me the resources to complete it.

Thanks also go out to my sister Liana VanOrmer who was always willing to watch my daughter in order for me to complete the requirements for this degree. Without her assistance, I may never have been able to fully dedicate myself to this program and be as successful as I was. Your continual support throughout is greatly appreciated!

Finally, I owe many thanks to the wonderful faculty and staff at Dakota State University for their support and encouragement throughout this program. You are always willing to go that extra mile to ensure our comprehension and success! Thank you so much for all you do!

ABSTRACT

The composition of student's today attending a university varies greatly. USDSU has a very unique student-base comprised mainly of non-traditional students. The students are on the campus for limited amounts of time mainly to attend their courses. These same students may be in need of certain services or products and may be unaware of how to obtain the information. The Resource Center at USDSU does currently utilize the traditional corkboard bulletin board but many students are unaware of this bulletin board. By implementing this online/interactive bulletin board for USDSU, this allows for quicker disbursement of information to the students as well as a convenient means for accessing the information.

DECLARATION

I hereby certify that this project constitutes my own product, that where the language of others is set forth, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions or writings of another.

I declare that the project describes original work that has not previously been presented for the award of any other degree of any institution.

Signed,

<Student name>

TABLE OF CONTENTS

APPENDICES

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

Background of the Problem

USDSU began in 1992 as the Sioux Falls Center for Higher Education. The initial location was the Bergeland Center, 132 S. Dakota Ave in downtown Sioux Falls which consisted of South Dakota State University and the University of South Dakota offices and seven classrooms. Dakota State University offices and classrooms were located at 3534 S. Western Ave. In December of 1997, Dakota State University offices and computer labs relocated from the previous location to the first floor of the Bergeland Center. (FY2005 Report)

In the Fall of 2001, the Sioux Falls Center for Higher education became USDSU (USD.SDSU.DSU). The institution also relocated to a leased building on the Southeast Technical Institute campus in response to increased class enrollment and course offerings. Within the first year at the new location, USDSU enrolled 1,579 students completing 10,297 credit hours. This represented a 48% increase in credit hours delivered in the first year in the new location. (FY2005 Report)

USDSU is a collaboration of four outstanding state public universities within South Dakota - the University of South Dakota, South Dakota State University, Dakota State University, and most recently Northern State University. These four schools have unique and competitive programs that are offered on the USDSU campus; it is an innovative model for higher education delivery.

USDSU enrollment numbers continue to increase on a yearly basis. According to statistics obtained from the 2004 – 2005 SD Board of Regent's Accountability Report, "USDSU at Sioux Falls served 3,372 students in three academic semesters" (SD Board of

Regents, 2005). USDSU also experienced a 3.2 percent growth in academic credit hours offered to students within that same time frame. Figures 2 and 3 illustrate the increase in headcount at USDSU each successive semester from Fall 01 to Fall 04 as well as Spring 02 to Spring 05. The most significant increases occurred within the timeframe Fall 01 – Fall 02 and Spring 02 to Spring 03. Enrollment increased by 240 in the Fall 01 – Fall 02 timeframe and 198 in the Spring 02 – Spring 03 timeframe. The next academic year enrollment increased by 258 students overall. In the final academic year (Fall 03 – Fall and Spring 04 – Spring 05), enrollment increased by 108 students.

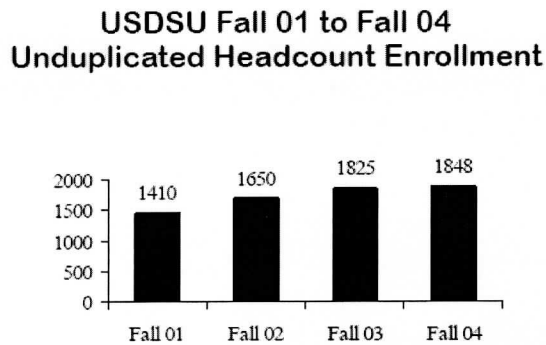


Figure 1. Fall 01 to Fall 04 Enrollment

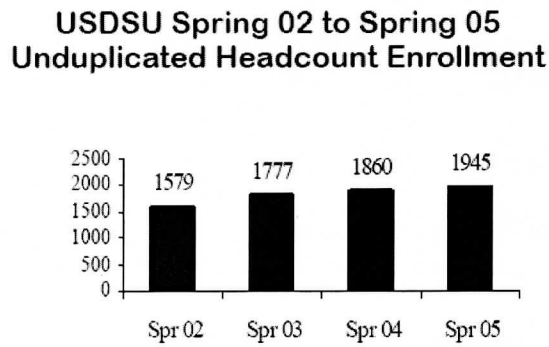


Figure 2. Spring 02 to Spring 05 Enrollment

Higher education institutions are situated in many different communities throughout South Dakota. In order to ensure continued success, USDSU makes available programs and

services that meet the needs of the growing community. The institutitional structure must meet varying needs on a variety of students who attend at our facility.

USDSU is very unique in the fact that the student base is comprised of mainly non-traditional students. Table 1 indicates the age group breakdown of our student base. A majority of our students are between the ages of 18 and 29 and have varied backgrounds.

Table 1. Age Group Breakdown of USDSU

AGE GROUP		
AGE GROUP	NUMBER OF STUDENTS	PERCENTAGE
17 or younger	4	0.2%
18 - 23	751	40.9%
24 - 29	551	30.0%
30 - 39	319	17.4%
40 - 49	157	8.6%
50 or older	54	2.9%
TOTALS	1836	100%

Many students have commitments outside of their education that impact their academic choices. Time is a critical issue to them, they are limited as to how much of their time can be spent on our campus locating and receiving the services they may need. There is currently a lack of online resources available for the current student base as well as prospective students in regards to the student services area.

On a traditional campus similar to Dakota State, located in Madison, South Dakota, freshman/sophomore housing is available, and required of new students. At USDSU, the students are not required to live on campus due to the fact that USDSU does not have student housing. Instead, the student population must locate apartments or alternative housing on their own. Some students are unfamiliar with the Sioux Falls metro area and are unsure where to

begin searching for housing. In some cases, the single unit housing living is not affordable and they are in need of another individual to split housing costs.

USDSU does not offer traditional housing such as other college campuses. Since USDSU do not offer housing to its students, USDSU is in need of an alternative mechanisms for the dissemination of with regard to not only housing, but many other areas.

Many students can complete their full degree on the USDSU campus, but there are a few students that may be required to complete some of their courses on one of the main campuses due to limited course offerings. There are other students who choose to complete their general education requirements or prerequisites for certain programs and upon completion commute to the corresponding main campus. Ride sharing has become more commonplace due to the commuting required by our students.

Many students may not be aware that there are USDSU students commuting to main campus on the same days as they may be with whom they could ride share with. This bulletin board would provide an electronic means to propagate this information to make commuters more aware of the possibility of ride-sharing with other individuals. This could reduce monthly gas expense as well as vehicle maintenance. With our harsh South Dakota winters, this would also provide them with another individual to be in the vehicle during the hazardous commute. Allowing students the ability to post ride sharing ads will allow them to reach more potential individuals to car pool with that they may never have from the Sioux Falls campus.

Childcare is another issue with the non-traditional students attending USDSU. Daycares are difficult to locate in Sioux Falls, especially for students who may be new to the Sioux Falls area. A majority of USDSU students work during the day and attend night courses. There has also been an increase in single parents attending USDSU campus. There is

limited evening childcare availability and this bulletin board would allow students to place childcare ads on the web site when they are in need of this service. This will allow them to reach a much broader audience and increase the possibility of locating childcare more promptly.

Many students leave textbooks, cell phones, and other items in classrooms. In most cases the item can be turned into the front office. The exception occurs with evening classes that begin at 7:00 p.m. and do not conclude until 9:45 p.m. The offices close at 7:00 p.m., so the office is not open for the individual to turn in lost and found items. By incorporating a lost and found, these students can post the lost/found item on the website to more quickly return the item to the rightful owner. This will allow them the opportunity to get the word out to a larger amount of people and will increase the possibility of the item being returned to the original owner.

USDSU currently utilizes a traditional bulletin board to distribute ads for such areas as housing, ride sharing, and childcare. There is limited space with a traditional bulletin board that could be alleviated with implementation of the online bulletin board. The location of the bulletin board is also an issue. The board is currently located in our Resource Center on our campus and many students do not frequent this area and the target audience may not be receiving the message.

Statement of the Problem

As previously stated, the majority of USDSU's current student population base is non-traditional. Many of our students may take one or two courses, depending on their circumstances. Table 2 represents a breakdown of how many courses each student is taking. The highest percentage indicated within this table is students that take only one class per

semester on our campus. This means they may be on our campus only one or two days a week.

Table 2. Number of Courses Taken Per Student at USDSU

NUMBER OF COURSES TAKEN		
NUMBER OF COURSES	NUMBER OF STUDENTS	PERCENTAGE
1	701	38.2%
2	432	23.5%
3	190	10.4%
4	302	16.5%
5	98	5.3%
6	17	0.9%
7	38	2.1%
8 or more	58	3.2%
TOTALS	1836	100%

Many of these same students have jobs and families outside of school which occupies much of their time. When they come to campus to attend their courses, they arrive just in time for the beginning of class and leave immediately after the conclusion of their course. If their course is located on the first floor of the building, they may not even be aware where the Resource Center is located, or even that it exists.

The fastest and easiest method of circulating this information to students is through an online bulletin board. The bulletin board would allow students to post ads online and reach many more students than a traditional in-house bulletin board. Since the ads will be online, it is more accessible to the students and allows the individual posting the ad to reach more students.

With this bulletin board implemented, students can now post messages and interact online from any location with Internet access. The online/interactive bulletin board consists of

six different categories to best meet student needs including housing, used textbooks, ride sharing, lost and found, childcare, and miscellaneous.

Objectives of the Project

The main objective of this site is providing an alternative method of disseminating information to the USDSU student population which will assist them in locating the services they need. In turn, individuals viewing the postings will also be able to respond to the posts and provide the service or locate an item that they may not have previously been aware of. The main individuals who will utilize this board will be USDSU students, faculty, and staff members. This will provide a quick and efficient means of allowing dissemination of information.

The deliverable for the project is a multi-page online bulletin board that allows for submission from individuals that will be posted to the website corresponding to the subject (textbooks for sale, ride sharing, lost and found, and housing). This information will be included as part of USDSU's current website for the students, faculty, staff, and other community members to observe.

Two different Access databases were created, one to handle the bulletin board postings and the other to store administrative usernames and passwords for any individuals within the organization that will maintain the bulletin board. If the individuals who post the messages need to go back and edit their postings, they can access their posting by signing in with the username and password stored in the central database. The creator also has the ability to delete their own posting (only after providing their credentials). The board will not allow any unsuitable material. All postings will be diligently screened and approved by an appointed individual at USDSU.

The fully functional website includes an initial screen (figure 3) that includes links to the following pages:

a. A page that explains how to navigate the bulletin board (howtopost.htm)

b. A page that allows the user to view all postings (viewAllPostMaster.asp)

c. A page that allows the user to search the postings (search.asp)

d. A page for each category that displays only the listings matching that particular category (childcareMaster.asp, miscMaster.asp, rideshareMaster.asp, textbookMaster.asp, and rentalMaster.asp)

e. A page that allows the user to create a new posting (createPost.asp)

f. A page that allows the user editing and deleting capabilities (updatePosting.asp)

g. A sign-in page for the user who is attempting to edit their posting (studentLogin.asp)

h. A sign-in page for the user who is attempting to delete their posting (studentLogin2.asp)

i. A sign-in page for the administrative personnel (adminLogin.asp)

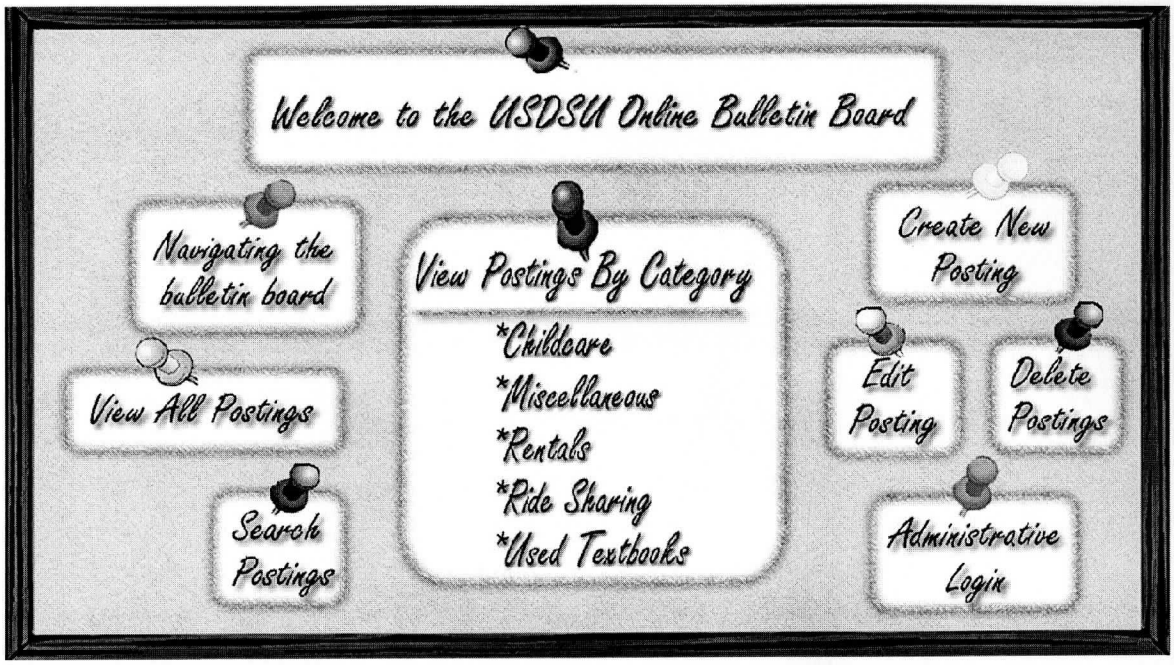


Figure 3. Initial Page (index.asp) of Bulletin Board

The user has easy access to the bulletin board and its many features. If they are unsure how to utilize the bulletin board, they can view the "Navigating the bulletin board" link. This page explains how to create new postings, edit postings, search postings, and view postings.

The user can create a new posting by simply clicking on that corresponding hot spot on the main page. If after creating the posting they realize that they made an error in their post, they can choose the "Edit Posting" hot spot. This will take them to a login page where they will be asked to input their username and password. Once they have inputted the correct information, the system pulls up their corresponding post and allows them to make and submit changes. The user also has the option of deleting their post altogether. They would instead click on the "Delete Posting" hotspot on the index.asp page. Again, this page requests credentials from the user and verifies their username and password. Once they have been verified, they then have the option of deleting their post. The post will not be available to view until the bulletin board administrator has approved the changes made to ensure no inappropriate material is allowed to be published on our website.

The user has many options for viewing the bulletin board postings. They can choose to "View All" postings. This choice allows them to view all postings on the bulletin board with no specific criteria and no particular order. The user can also choose to "View the postings by category." This allows the user to view only a specific category that may be of interest to them. Finally, they can "Search Postings" by clicking on the corresponding hot spot. This allows the user to provide a key word/phrase for the bulletin board to search for. The only postings displayed to the user are those that meet the search criteria.

DEFINITION OF TERMS

The following terms will have continued use throughout this study. The definitions will minimize the chance for confusion relative to these terms.

ASP - is Microsoft's server-side technology for dynamically-generated web pages that is marketed as an add-on to Internet Information Services (IIS). (Wikipedia.org).

administrator – is a person who is responsible for the environmental aspects of a database. (Wikipedia.org)

attribute - an abstraction of a characteristic of an entity or substance. (Wikipedia.org)

client-side cursor - in a client-side cursor, the server sends the entire result set across the network to the client computer. The client computer provides and manages the temporary resources needed by the cursor and result set. The client-side application can browse through the entire result set to determine which rows it requires. (MSDN Library)

fields - data that has several parts can be divided into fields. For example, a computer may represent today's date as three distinct fields: the day, the month and the year. (Wikipedia.org)

general education requirements – A group of basic courses for all students in a course of study at a college or university. Usually these courses cover broad subjects in the arts, social sciences, and natural sciences. (CollegeinColorado.org)

IIS - Microsoft Internet Information Services (IIS; sometimes, erroneously called Server or System) is a set of Internet-based services for servers using Microsoft Windows. (Wikipedia.org)

JavaScript - a scripting programming language based on the concept of prototypes. The language is best known for its use in websites, but is also used to enable scripting access to objects embedded in other applications. (Wikipedia.org)

Macromedia Dreamweaver - is a web development tool, created by Macromedia (now Adobe Systems), which is currently in version 8. Initial versions of the application served as simple WYSIWYG HTML editors but more recent versions have incorporated notable

support for many other web technologies such as CSS, JavaScript, and various server-side scripting frameworks. (Wikipedia.org)

Microsoft Access - is a relational database management system from Microsoft, packaged with Microsoft Office Professional which combines the relational Microsoft Jet Database Engine with a graphical user interface. It can use data stored in Access/Jet, SQL Server, Oracle, or any ODBC-compliant data container. Skilled software developers and data architects use it to develop powerful, complex application software. (Wikipedia.org)

non-traditional student - term referring to students at higher education institutions (undergraduate college or university) whom are not of the typical age as the majority of their peers. (Wikipedia.org).

prerequisites – A course required in preparation for another, usually more advanced course. (CollegeinColorado.org)

SQL - (Structured Query Language) is the most popular computer language used to create, modify and retrieve and manipulate data from relational database management systems. The language has evolved beyond its original purpose to support object-relational database management systems. (Wikipedia.org)

server-side cursor – With a server-side cursor, the server manages the result set using resources provided by the server computer. The server-side cursor returns only the requested data over the network. This type of cursor can sometimes provide better performance than the client-side cursor, especially in situations where excessive network traffic is a problem.

traditional student – term referring to students at higher education institutions (undergraduate college or university) whom are the typical age as the majority of their peers. (Wikipedia.org)

Windows - is a series of proprietary operating environments and operating systems created by Microsoft for use on personal computers and servers.

PROJECT SCOPE

The scope of the online/interactive bulletin board consisted of project initiation, project planning, project execution and project closing. Corresponding to the activities involved with all stages of my project, I have created a Gantt chart and Work Breakdown structure (see Appendix A & B).

The expected outcome of this project was an online/interactive bulletin board for USDSU students. Instead of posting messages on the traditional bulletin board in our Resource Center, they can access and post their own items for public viewing and reach far more individuals than the traditional posting method.

What this project offered was a unique solution to the traditional bulletin board. Instead of having to physically enter the building, the individual posts their ad online from the comfort of their own home. Individuals in need of the service are now provided the ability to view the postings from the privacy of their own home. They can also reply immediately via various contact methods including an automated response to the originator of the posting.

Technical Considerations

I had many variables to take into consideration in regards to this project. One of the key components was the technical considerations including the current technology utilized by the clients and available to me to complete the project as well as ease of use.

USDSU utilizes Microsoft Access for database creation, storage, and retrieval. It was logical that I would utilize this program to create my databases for this particular project. The current website designer uses JavaScript as well as ASP with Macromedia Dreamweaver as the web page editor of choice.

ASP utilizes a "template-based approach to server-side scripting" (Bakharia, 2002, p. 79). The script can be written in either Jscript or VBscript and is embedded within the web page. Another distinct advantage ASP offers over other languages is that is free (it is included with IIS) and it works very well in a Windows-based environment. USDSU operates a predominantly Windows-based environment. Another factor that influenced the decision is its ability to effectively communicate with a Microsoft Access database (Bakharia, 2002).

The server is already in place for this site, so that is one aspect that was already decided for me. IIS is utilized by our campus as the server for our website. This was advantageous when completing this project due to ASP's functionality with IIS. This was one less hurdle I had to tackle when implementing the web site (Bakharia, 2002).

The web page designer and I have access to both programs required in order to successfully complete the project. Due to the limited time allotted for completion and implementation of the project, the determination made was to use the existing software development platform consisting of Access and ASP. Other influential factors included our extensive knowledge and experience with JavaScript, ASP and Dreamweaver. We also needed to ensure that all components would work well and without too much difficulty.

SYSTEM DESIGN

USDSU has chosen to create and implement its own online/interactive bulletin board. There are many various "free" bulletin boards available to any organization. These types of bulletin boards are utilized by many organizations to avoid the hard-coding involved with designing and creating your own online bulletin board. A system such as this can be extremely complex in nature. There are many variables that must be taken into account when creating any online forum. Some of the variables that I had to take into consideration included privacy, server space, login requirements, database structure/access, and programming language to utilize.

Database Design

Two different Access databases were created, one to handle the bulletin board postings and the other to store administrative usernames and passwords for any individuals within the organization that will maintain the bulletin board. If the individuals who post the messages need to go back and edit their postings, they can access their posting by signing in with the username (email address) and password stored in the central database. The creator also has the ability to delete their own posting (only after providing their credentials). The board will not allow any unsuitable material. All postings will be diligently screened and approved by an appointed individual at USDSU.

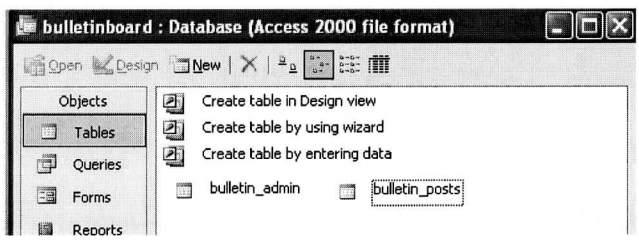


Figure 4. Bulletin Board Database

Figure 4 illustrates the overall Microsoft Access database created for the bulletin board. The database contains two tables named bulletin_admin and bulletin_post. The bulletin_admin table can be viewed in figure 5 and consisted of the following fields and attributes:

a. Key – autonumber data type, long integer, default value of zero, indexed

b. Username – text data type, maximum field size of fifteen

c. Password – text data type, maximum field size of ten

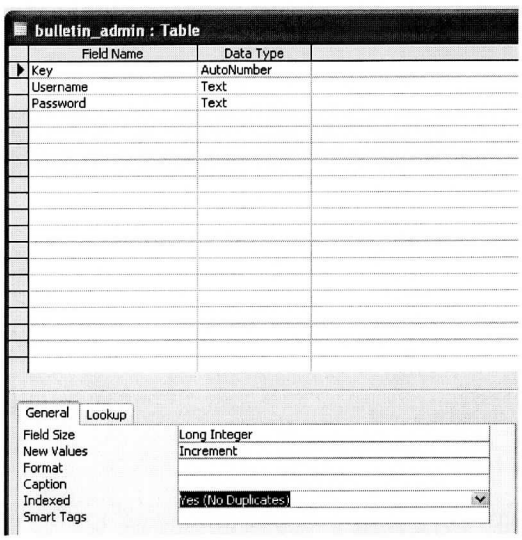


Figure 5. bulletin_admin Table

Figure 6 depicts the current contents of the bulletin_admin table. The only person able to add or remove users is the administrator. This particular aspect of the bulletin board is not connected to the web site. The administrator will manually enter in any individual allowing them administrative rights to the bulletin board.

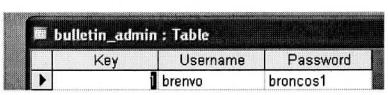


Figure 6: bulletin_admin Table Contents

The other table associated with the bulletin board database, bulletin_posts, is depicted in figure 7. The table is used in many different aspects of the overall bulletin board. The title, category, name, email, date_of_post, and description fields are all utilized in the search.asp page, viewAllPosts.asp, studentLogin.asp, updatePosting.asp, childcareMaster.asp, miscMaster.asp, rentalMaster.asp, rideshareMaster.asp, and textbook.asp. This table consisted of the following fields and attributes:

a. Key – autonumber, long integer, indexed
b. Title – text data type, maximum twenty-five characters in length
c. Category – text data type, maximum fifteen characters in length
d. Name - text data type, maximum thirty characters in length
e. Email/Confirm Email - text data type, maximum twenty-five characters in length
f. Password/Confirm Password - text data type, maximum ten characters in length
g. Date_of_Post – date/time data type
h. Description – text data type, maximum 125 characters
i. Approval_of_Post – yes/no data type, default value is "no"

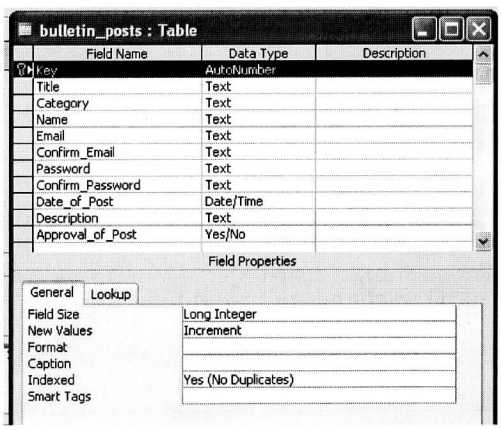
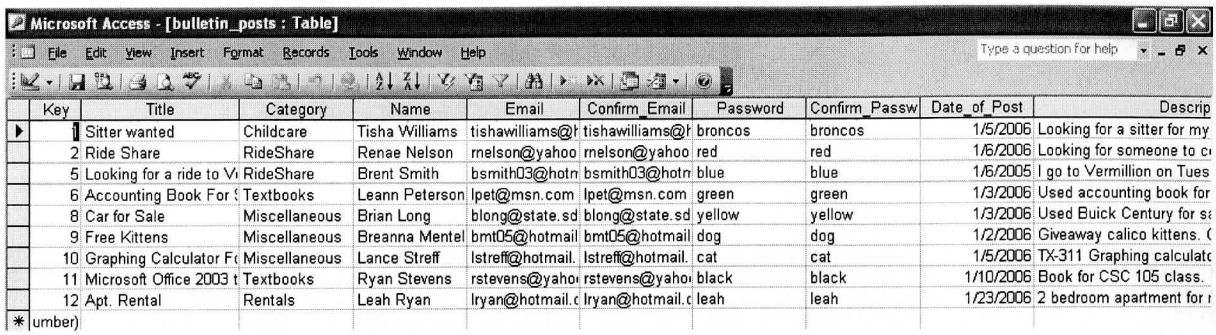


Figure 7. bulletin_post Table

Figure 8 depicts the current contents of the bulletin_post table. Each of the listings is written from the "Create Post" web page on the bulletin board. A connection was established between this particular database and the createPost.asp page. When the user has completed all areas of the createPost.asp page and mouse clicks on the submit button, the record is added to the database.



Key	Title	Category	Name	Email	Confirm_Email	Password	Confirm_Passw	Date_of_Post	Descrip
1	Sitter wanted	Childcare	Tisha Williams	tishawilliams@h	tishawilliams@h	broncos	broncos	1/5/2006	Looking for a sitter for my
2	Ride Share	RideShare	Renae Nelson	rnelson@yahoo	rnelson@yahoo	red	red	1/6/2006	Looking for someone to c
5	Looking for a ride to V	RideShare	Brent Smith	bsmith03@hotm	bsmith03@hotm	blue	blue	1/6/2005	I go to Vermillion on Tues
6	Accounting Book For S	Textbooks	Leann Peterson	lpet@msn.com	lpet@msn.com	green	green	1/3/2006	Used accounting book for
8	Car for Sale	Miscellaneous	Brian Long	blong@state.sd	blong@state.sd	yellow	yellow	1/3/2006	Used Buick Century for s
9	Free Kittens	Miscellaneous	Breanna Mentel	bmt05@hotmail	bmt05@hotmail	dog	dog	1/2/2006	Giveaway calico kittens. (
10	Graphing Calculator Fo	Miscellaneous	Lance Streff	lstreff@hotmail.	lstreff@hotmail.	cat	cat	1/5/2006	TX-311 Graphing calculat
11	Microsoft Office 2003 t	Textbooks	Ryan Stevens	rstevens@yahoo	rstevens@yahoo	black	black	1/10/2006	Book for CSC 105 class.
12	Apt. Rental	Rentals	Leah Ryan	lryan@hotmail.c	lryan@hotmail.c	leah	leah	1/23/2006	2 bedroom apartment for
umber)									

Figure 8. bulletin_post Table Contents

If the user returns to edit or delete their particular posting, the studentLogin.asp page will verify their identity. Figure 9 depicts the studentLogin.asp and the studentLogin2.asp page. The page consists of a form, two textboxes, and a submit button. When the user enters the information into the username and password and presses the submit button, the contents of the textboxes are compared to the email and password fields in the bulletin_post table.

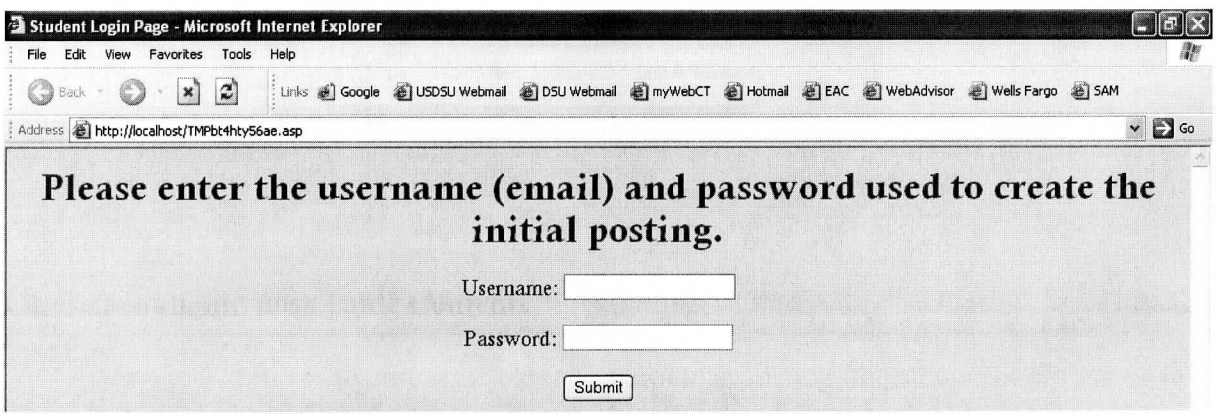


Figure 9. studentLogin.asp and studentLogin2.asp

If there is a match, the corresponding posting is displayed to the user. They can then choose to either edit or delete their posting from this page. If they choose to update or edit their posting, the new information is written back to the bulletin_post table, replacing the previous information. If the user chooses to delete their posting, the posting is removed from the bulletin_post table.

Figure 10 illustrates the adminLogin.asp page of the online bulletin board. Again, a form, two textboxes, and a submit button are included. When the form is submitted, the information entered is then compared against the current contents (username and password) within the bulletin_admin table.

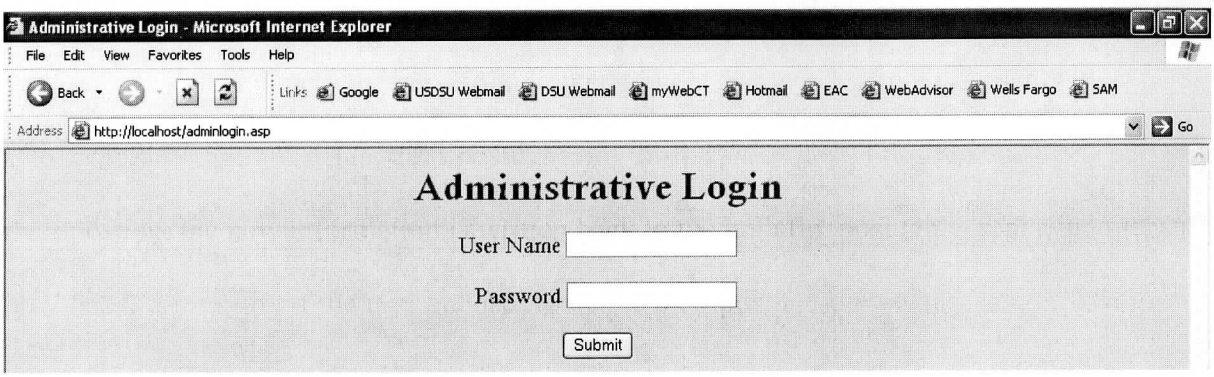


Figure 10. adminLogin.asp

If the user is validated, they have administrative access. They are then able to view all new postings and have the choice to approve or disapprove the postings. Figure 11 illustrates the page that is displayed to the administrator.

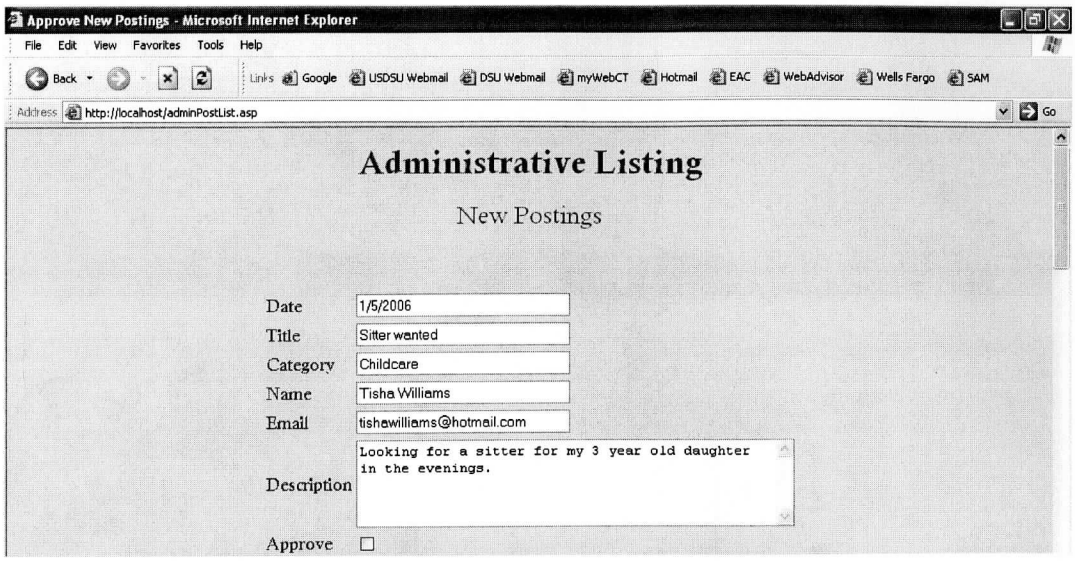


Figure 11. adminPostList.asp

The fields displayed in the administrative listing are date, title, category, name, email and description along with a checkbox for the administrator to either approvied or reject the posting. If the posting is rejected, it will not be allowed on our bulletin board for students to view. If the posting is approved, the postings will be available to view by students.

Web-interface Design

The first aspect I had to consider when creating the web-interface design was which scripting mechanism should be implemented. Server-side was used for a majority of the site, along with a few client-side scripts. My overall goal for this site was to make it visually pleasing to the user as well as easy to navigate.

More bells and whistles certainly could have been included, but my main goal was to provide an effective means of communicating with the students while ensuring that color scheme and other design aspects coincides with the current USDSU web site which can be viewed in figure 12. Since USDSU already has a general website in place, the bulletin board needed to follow the overall design schematic.

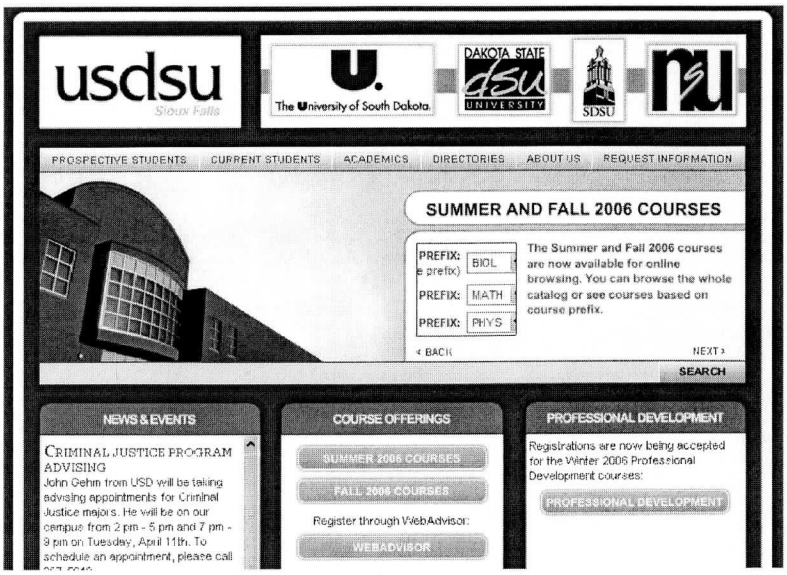


Figure 12. USDSU Home Page

I did first consider utilizing frames for the online bulletin board. I decided against implementing frames after considering that the site will be placed on the USDSU main site. The webmaster already has a very distinctive layout (with frames). Adding a web page with frames inside a webpage that already utilizes frames did not seem like a good design schematic for the USDSU online bulletin board.

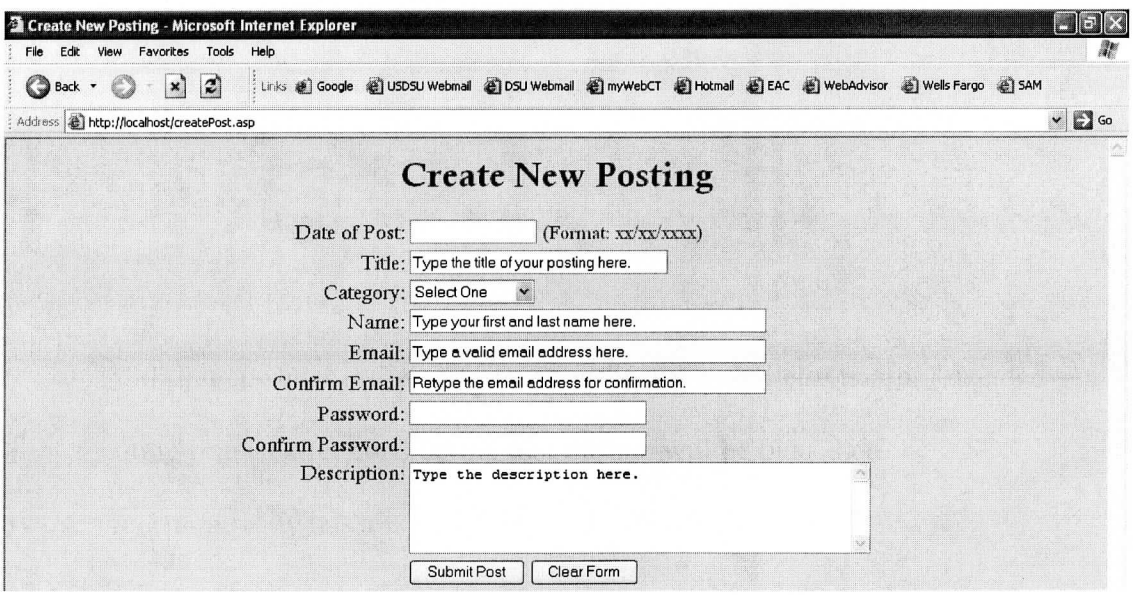


Figure 13. createPost.asp

The first and most important aspect of this bulletin board is creating a new post. Figure 13 shows the Create New Posting page that is displayed to the user when selecting "Create New Post" from index.asp. Each posting includes the following: date, title, category, (with a drop-down box), name, email, confirm email, password, confirm password, and description.

After entering the information, the form validates the information when the user mouse clicks on the "Submit Post" button. If all areas are not complete, the user will receive an alert window similar to figure 14. The alert window will display the particular errors the user made if they left fields empty or null or if they did not enter the information in the required format.

Figure 14. Error Message

The user also has the option here to "Clear Form" if they are not satisfied with what they have entered or if they choose not to post at that time. Once the user has corrected their mistakes and has successfully submitted the post, they will receive a confirmation similar to figure 15. The confirmation consists of a text confirmation that the submission has been received as well as a linked hover button that allows them to return to the home page from the current page. As discussed previously, the user can edit and delete their postings at a later date by providing their credentials at the studentLogin.asp site.

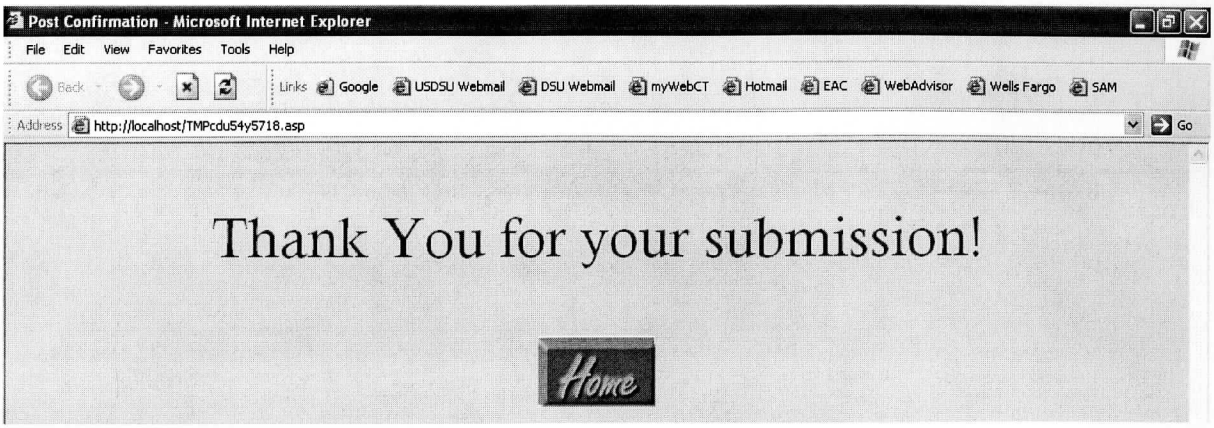


Figure 15. confirmPost.asp

Users can view the various postings included on the board in a number of ways. The first option available to them is to view all postings. Figure 16 shows what information is displayed to the user when they choose to view all postings from the main page.

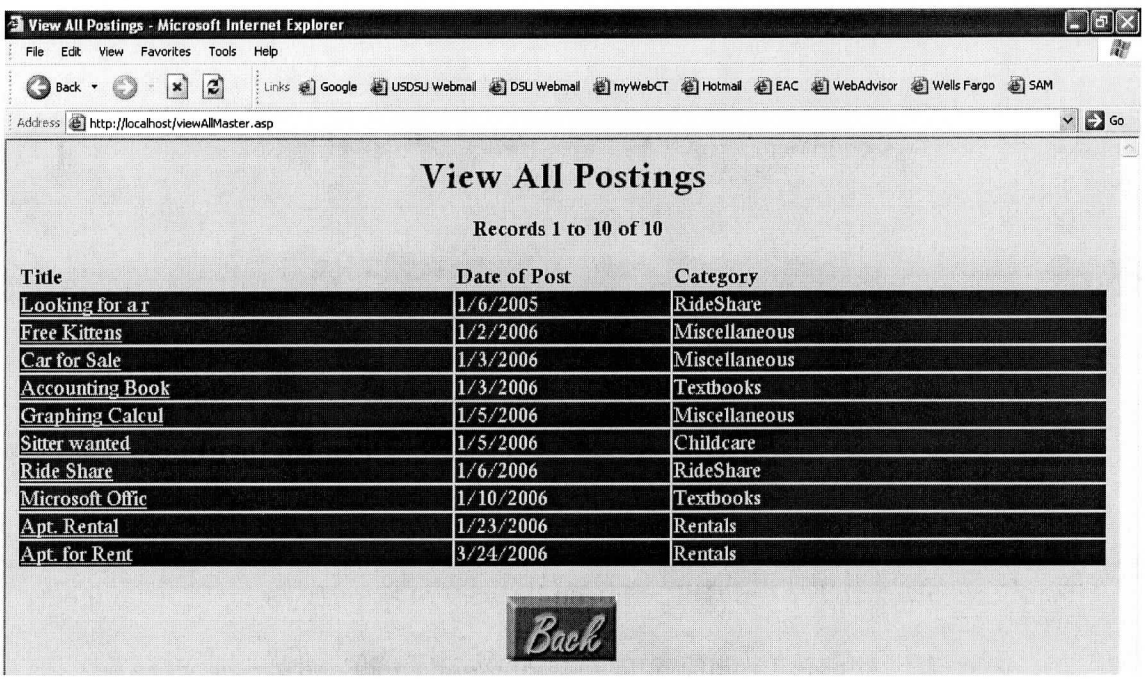


Figure 16: viewAllMaster.asp

This figure displays what record numbers are being viewed. It is set up to display ten postings at a time. This view also contains a brief listing of all posts currently available

including the title, date of the post, and the category. The user also has the option of mouse clicking the title of each post in order to view the more detailed version. Figure 17 shows the detailed view of a particular post. In this view, the user can view the date of the post, title, the name of the individual who created the posting, the email of the contact person, and the description. This view also includes a hover back button that allows the user the option to go back to the previous master view of the listing.

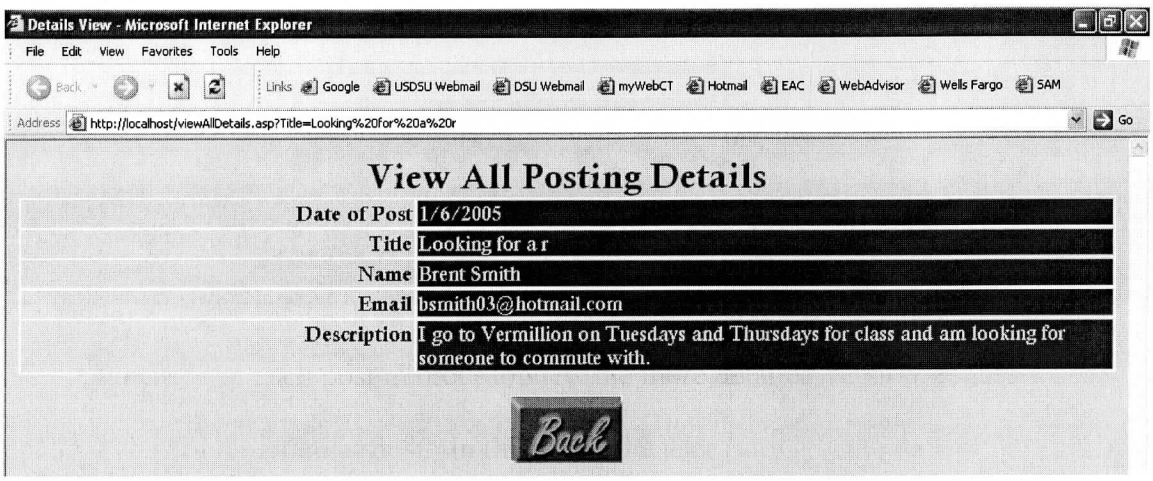


Figure 17: viewAllDetail.asp

The next method that can be utilized by users to view the postings is via a search engine. The user can choose to search the current postings to locate any postings that may be of interest to them.

Figure 18: search.asp

Figure 18 shows the search.asp page which includes a textbox for the user to enter in their specific search term. A submit button is also included for the user to submit the search query.

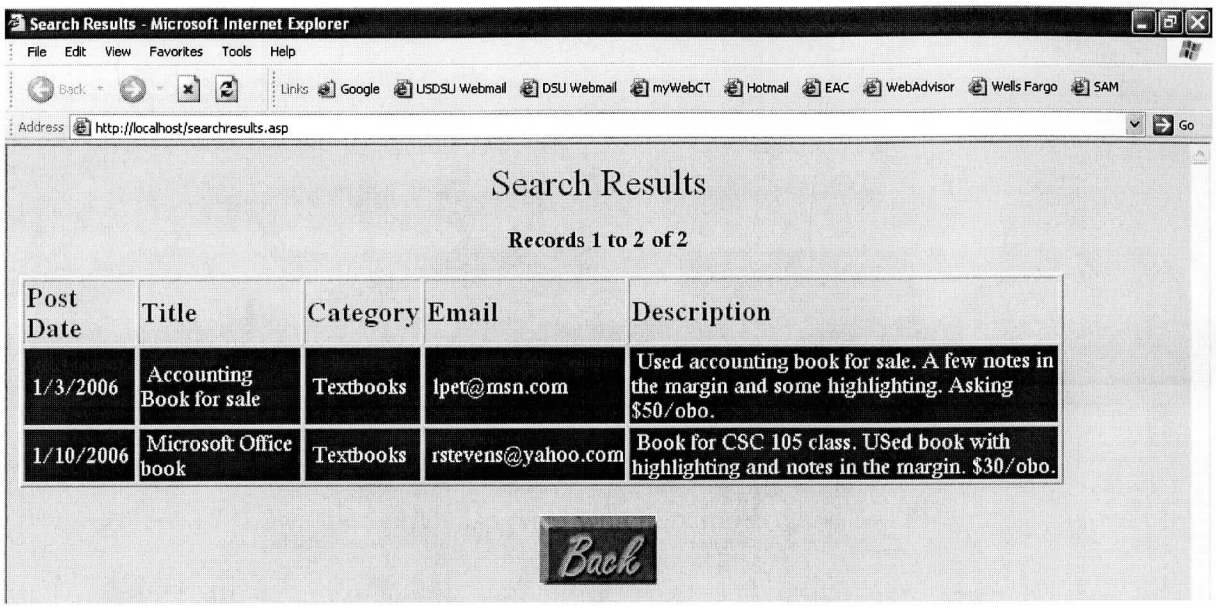


Figure 19: searchResults.asp

Figure 19 is the searchResults.asp page that is displayed after the user mouse clicks the submit button. The text from the textbox is compared to the bulletin board postings in the bulletin_post table in the database. In figure 19, all of the records meeting the search criteria of "book" are displayed to the user. Depending upon the search term/phrase, the query returns varying results.

The final method for viewing the bulletin board posts is a division of the postings by category. The index.asp page allows the users the ability to choose a particular category to view only the corresponding postings to the particular category. The user has the choice of viewing the postings by childcare, miscellaneous, rentals, rideshare, or textbooks. The user can also view a more detailed version of the post they are interested in by mouse clicking the

title of the post. An SQL query is utilized to retrieve the corresponding information for each category.

The following is the SQL code utilized to display only postings within the **childcare** category:

```
<%
Dim OrderByChildcare
Dim OrderByChildcare_numRows
Set OrderByChildcare = Server.CreateObject("ADODB.Recordset")
OrderByChildcare.ActiveConnection = MM_Bulletin_Board_STRING
OrderByChildcare.Source = "SELECT Title, Category, Name, Email, Date_of_Post,
Description  FROM bulletin_posts  WHERE Category LIKE 'Childcare'  ORDER BY
Date_of_Post"
OrderByChildcare.CursorType = 0
OrderByChildcare.CursorLocation = 2
OrderByChildcare.LockType = 1
OrderByChildcare.Open()
OrderByChildcare_numRows = 0
%>
```

The DIM statements in the first two lines of code are used to declare the variables "OrderByChildcare" and OrderByChildcare_numRows". The Set OrderByChildcare = Server.CreateObject("ADODB.Recordset") is used to be able to read database data. Before the data can be read, it must first be loaded into a recordset which is done by the ADODB.Recordset statement in the code. The code OrderByChildcare.ActiveConnection = MM_Bulletin_Board_STRING indicates to which connection object the specified recordset object currently belongs. In this case, the connection is being made to the bulletin_board database.

The code includes SQL statements that are utilized to narrow the displayed results. The SELECT statement indicates which fields from a particular table to display. The fields that have been "selected" from the bulletin_post table are Title, Category, Name, Email, and

Date_of_Post. The next statement in the query is the word FROM. This statement points to the table to pull the information "from." In this statement, the table the information is being taken from is the bulletin_posts table. The next portion of the statement is LIKE. The LIKE statement locates only records that match the particular field name; in this case we are displaying "childcare" records. The final portion of the statement is ORDER BY. This particular statement determines how the records that meet the criteria are ordered. In this particular statement, the results are ordered by the Date_of_Post field.

The cursor type and lock type properties specified when opening a recordset is important. In the code OrderByChildcare.CursorType = 0 the cursor type is set to the default zero. What this means when the code is executed is that this moves forward through the records. The reason the forward only method is utilized when moving through the records is for performance. This type of cursor moves more speedily through the records and takes less time to complete.

The lock type only allows data to be read from, but not altered. In this particular case, I want the user to be able to read, but not write to the records. In the code "OrderByChildcare.LockType = 1", I am setting the locktype for this query to the value one which specifies a "read-only" attribute for this property. The cursor location property defines where the recordset is created when opened. In this line of code "OrderByChildcare.CursorLocation = 2", the cursor location is set to two which indicates that the recordset is created on the server-side cursor as opposed to the client-side cursor.

After this step of the code has completed, the next step is to OPEN the records and display them to the user. The Open method opens the database to allow access to records in a table, the results of a query, or to a saved Recordset. In this case, the only records that will be

displayed are the ones that fall under the "childcare" category. The final portion of the code is "OrderByChildcare_numRows = 0".

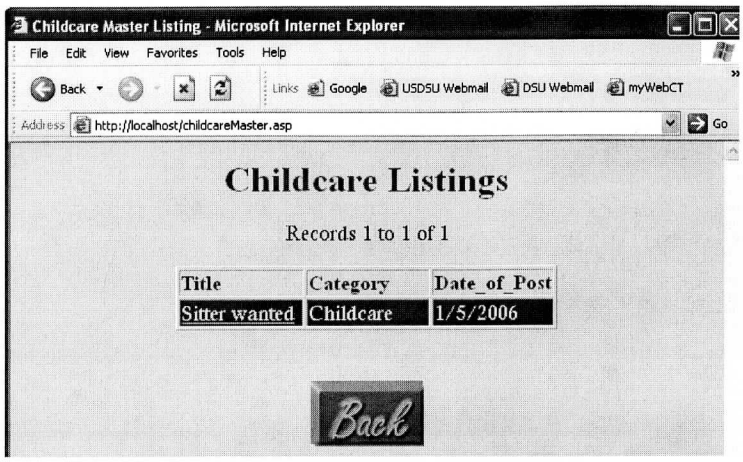


Figure 20: childcareMaster.asp

Figure 20, childcareMaster.asp, illustrates what is displayed to the user when mouse clicking on the childcare category. Again, the title, category, and date of post are displayed to the user. If the user would like to return back to the main page, they can click on the back button. If the user would like to view more details on a specific listing, they can click on the corresponding title to go to the detailed view.

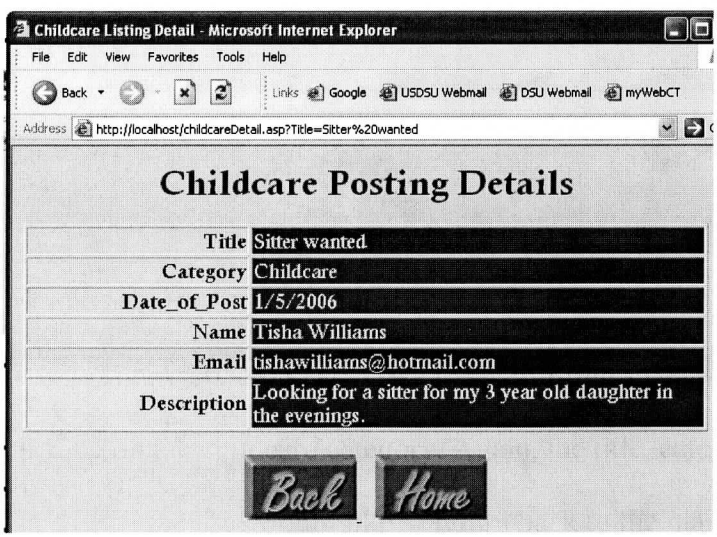


Figure 21: childcareDetail.asp

Figure 21 shows childcareDetail.asp, the detailed view for the user. The fields included in this view are title, category, name, email, and description. There are also two buttons – back and home for the user to easily navigate through the pages.

The following is the SQL code utilized to display only postings within the **miscellaneous** category:

```
<%
Dim Miscellaneous
Dim Miscellaneous_numRows
Set Miscellaneous = Server.CreateObject("ADODB.Recordset")
Miscellaneous.ActiveConnection = MM_Bulletin_Board_STRING
Miscellaneous.Source = "SELECT Title, Category, Date_of_Post  FROM
bulletin_posts  WHERE Category LIKE 'Miscellaneous'"
Miscellaneous.CursorType = 0
Miscellaneous.CursorLocation = 2
Miscellaneous.LockType = 1
Miscellaneous.Open()
Miscellaneous_numRows = 0
%>
```

The DIM statements in the first two lines of code are used to declare the variables "Miscellaneous" and "Miscellaneous_numRows Set Miscellaneous = Server.CreateObject("ADODB.Recordset") is used to be able to read database data. Before the data can be read, it must first be loaded into a recordset which is done by the ADODB.Recordset statement in the code. The code Miscellaneous.ActiveConnection = MM_Bulletin_Board_STRING indicates to which connection object the specified recordset object currently belongs. In this case, the connection is being made to the bulletin_board database.

The code includes SQL statements that are utilized to narrow the displayed results. The SELECT statement indicates which fields from a particular table to display. The fields that have been "selected" from the bulletin_post table are Title, Category, Name, Email, and Date_of_Post. The next statement in the query is the word FROM. This statement points to the table to pull the information "from." In this statement, the table the information is being taken from is the bulletin_posts table. The next portion of the statement is LIKE. The LIKE statement locates only records that match the particular field name; in this case we are displaying records from the "miscellaneous" category. The final portion of the statement is ORDER BY. This particular statement determines how the records that meet the criteria are ordered. In this particular statement, the results are ordered by the Date_of_Post field.

The cursor type and lock type properties specified when opening a recordset is important. In the code Miscellaneous.CursorType = 0 the cursor type is set to the default zero. What this means when the code is executed is that this moves forward through the records. The reason the forward only method is utilized when moving through the records is for performance. This type of cursor moves more speedily through the records and takes less time to complete.

The lock type only allows data to be read from, but not altered. In this particular case, I want the user to be able to read, but not write to the records. In the code "Miscellaneous.LockType = 1", I am setting the locktype for this query to the value one which specifies a "read-only" attribute for this property. The cursor location property defines where the recordset is created when opened. In this line of code "Miscellaneous.CursorLocation = 2", the cursor location is set to two which indicates that the recordset is created on the server-side cursor as opposed to the client-side cursor.

After this step of the code has completed, the next step is to OPEN the records and display them to the user. The Open method opens the database to allow access to records in a table, the results of a query, or to a saved Recordset. In this case, the only records that will be displayed are the ones that fall under the "miscellaneous" category. The final portion of the code is "Miscellaneous_numRows = 0".

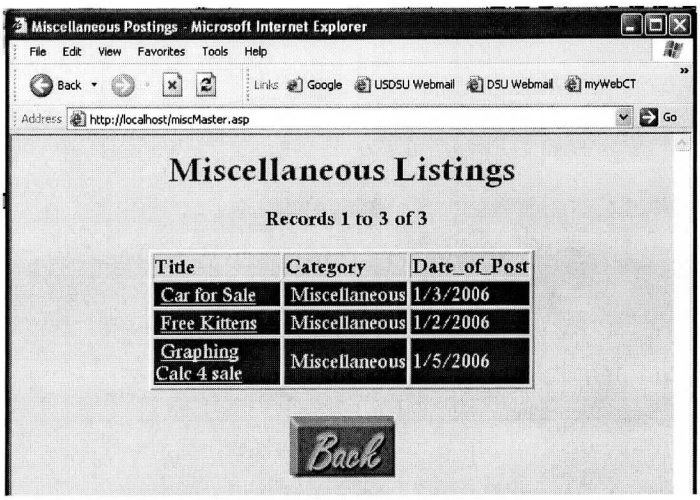


Figure 22: miscMaster.asp

Figure 22 illustrates what is displayed to the user when mouse clicking on the miscellaneous category. Again, the title, category, and date of post are displayed to the user. If the user would like to return back to the main page, they can click on the back button. If the user would like to view more details on a specific listing, they can click on the corresponding title to go to the detailed view.

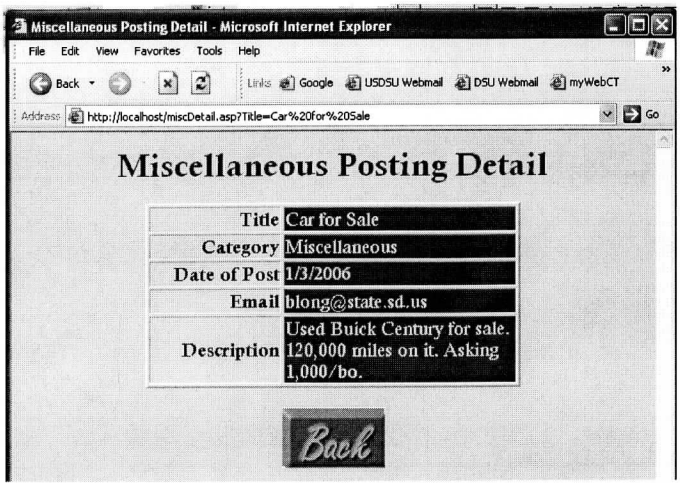


Figure 23: miscDetail.asp

Figure 23 displays miscDetail.asp, the detailed view for the user. The fields included in this view are title, category, name, email, and description. There is also a back button available which allows the user to easily navigate through the pages. The email is listed for the user to contact the individual who originally posted the message.

The following is the SQL code utilized to display only postings within the **rental** category:

```
<%
Dim Rental
Dim Rental_numRows
Set Rental = Server.CreateObject("ADODB.Recordset")
Rental.ActiveConnection = MM_Bulletin_Board_STRING
Rental.Source = "SELECT Title, Category, Date_of_Post  FROM bulletin_posts
WHERE Category LIKE 'Rentals'"
Rental.CursorType = 0
Rental.CursorLocation = 2
Rental.LockType = 1
Rental.Open()
Rental_numRows = 0
%>
```

The DIM statements in the first two lines of code are used to declare the variables "Rental" and "Rental _numRows. Set Rental = Server.CreateObject("ADODB.Recordset") is used to be able to read database data. Before the data can be read, it must first be loaded into a recordset which is done by the ADODB.Recordset statement in the code. The code Rental.ActiveConnection = MM_Bulletin_Board_STRING indicates to which connection object the specified recordset object currently belongs. In this case, the connection is being made to the bulletin_board database.

The code includes SQL statements that are utilized to narrow the displayed results. The SELECT statement indicates which fields from a particular table to display. The fields that have been "selected" from the bulletin_post table are Title, Category, Name, Email, and Date_of_Post. The next statement in the query is the word FROM. This statement points to the table to pull the information "from." In this statement, the table the information is being taken from is the bulletin_posts table. The next portion of the statement is LIKE. The LIKE statement locates only records that match the particular field name; in this case we are displaying records from the "rental" category. The final portion of the statement is ORDER BY. This particular statement determines how the records that meet the criteria are ordered. In this particular statement, the results are ordered by the Date_of_Post field.

The cursor type and lock type properties specified when opening a recordset is important. In the code Rental.CursorType = 0 the cursor type is set to the default zero. What this means when the code is executed is that this moves forward through the records. The reason the forward only method is utilized when moving through the records is for performance. This type of cursor moves more speedily through the records and takes less time to complete.

The lock type only allows data to be read from, but not altered. In this particular case, I want the user to be able to read, but not write to the records. In the code "Rental.LockType = 1", I am setting the locktype for this query to the value one which specifies a "read-only" attribute for this property. The cursor location property defines where the recordset is created when opened. In this line of code "Rental.CursorLocation = 2", the cursor location is set to two which indicates that the recordset is created on the server-side cursor as opposed to the client-side cursor.

After this step of the code has completed, the next step is to OPEN the records and display them to the user. The Open method opens the database to allow access to records in a table, the results of a query, or to a saved Recordset. In this case, the only records that will be displayed are the ones that fall under the "rental" category. The final portion of the code is "Rental _numRows = 0".

Figure 24 illustrates what is displayed to the user when mouse clicking on the rental category. Again, the title, category, and date of post are displayed to the user. If the user would like to return back to the main page, they can click on the back button. If the user would like to view more details on a specific listing, they can click on the corresponding title to go to the detailed view.

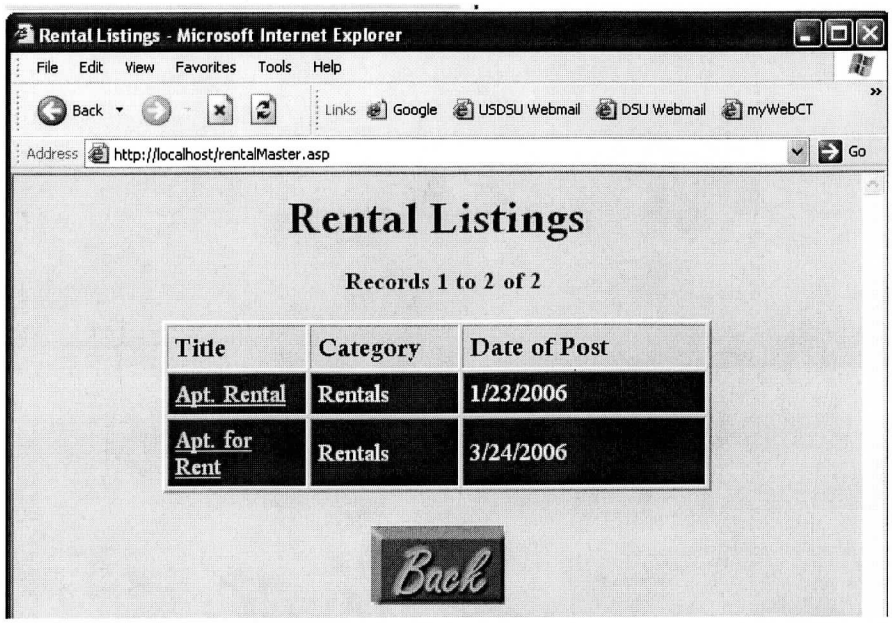


Figure 24: rentalMaster.asp

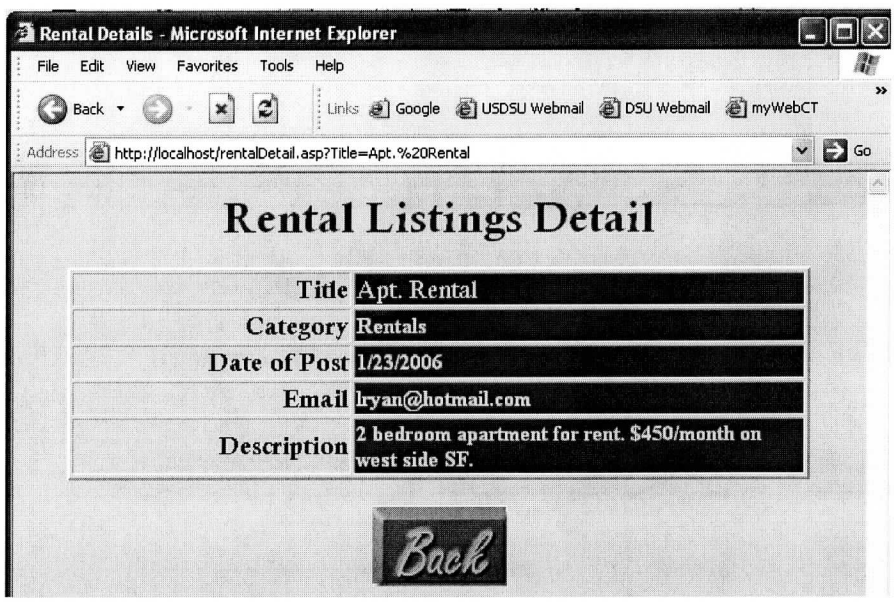


Figure 25: rentalDetail.asp

Figure 25 displays rentalDetail.asp, the detailed view for the user. The fields included in this view are title, category, name, email, and description. The user can easily locate the contact information in the detailed view in order to make the initial contact if interested in the item posted. There is also a back button for the user to easily navigate through the pages.

The following is the SQL code utilized to display only postings within the **rideshare** category:

```
<%
Dim rideShare
Dim rideShare_numRows
Set rideShare = Server.CreateObject("ADODB.Recordset")
rideShare.ActiveConnection = MM_Bulletin_Board_STRING
rideShare.Source = "SELECT Title, Category, Date_of_Post  FROM bulletin_posts
WHERE Category LIKE 'RideShare'"
rideShare.CursorType = 0
rideShare.CursorLocation = 2
rideShare.LockType = 1
rideShare.Open()
rideShare_numRows = 0
%>
```

The DIM statements in the first two lines of code are used to declare the variables "rideShare" and "rideShare _numRows. Set rideShare = Server.CreateObject("ADODB.Recordset") is used to be able to read database data. Before the data can be read, it must first be loaded into a recordset which is done by the ADODB.Recordset statement in the code. The code rideShare.ActiveConnection = MM_Bulletin_Board_STRING indicates to which connection object the specified recordset object currently belongs. In this case, the connection is being made to the bulletin_board database.

The code includes SQL statements that are utilized to narrow the displayed results. The SELECT statement indicates which fields from a particular table to display. The fields that have been "selected" from the bulletin_post table are Title, Category, Name, Email, and Date_of_Post. The next statement in the query is the word FROM. This statement points to

the table to pull the information "from." In this statement, the table the information is being taken from is the bulletin_posts table. The next portion of the statement is LIKE. The LIKE statement locates only records that match the particular field name; in this case we are displaying records from the "rideshare" category. The final portion of the statement is ORDER BY. This particular statement determines how the records that meet the criteria are ordered. In this particular statement, the results are ordered by the Date_of_Post field.

The cursor type and lock type properties specified when opening a recordset is important. In the code rideShare.CursorType = 0 the cursor type is set to the default zero. What this means when the code is executed is that this moves forward through the records. The reason the forward only method is utilized when moving through the records is for performance. This type of cursor moves more speedily through the records and takes less time to complete.

The lock type only allows data to be read from, but not altered. In this particular case, I want the user to be able to read, but not write to the records. In the code "rideShare.LockType = 1", I am setting the locktype for this query to the value one which specifies a "read-only" attribute for this property. The cursor location property defines where the recordset is created when opened. In this line of code "rideShare.CursorLocation = 2", the cursor location is set to two which indicates that the recordset is created on the server-side cursor as opposed to the client-side cursor.

After this step of the code has completed, the next step is to OPEN the records and display them to the user. The Open method opens the database to allow access to records in a table, the results of a query, or to a saved Recordset. In this case, the only records that will be

displayed are the ones that fall under the "rideshare" category. The final portion of the code is "rideShare _numRows = 0".

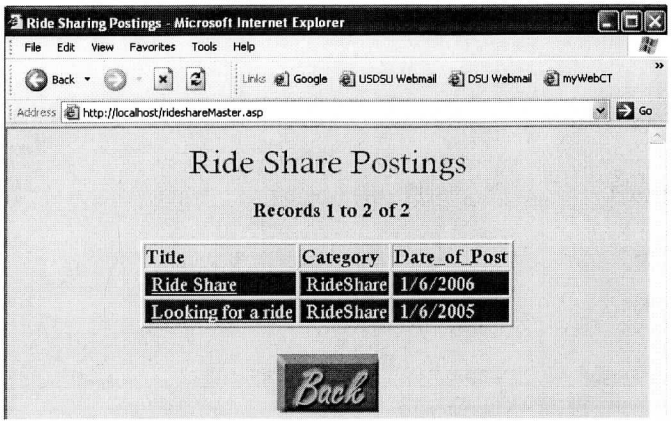


Figure 26: rideShareMaster.asp

Figure 26, rideShareMaster.asp, illustrates what is displayed to the user when mouse clicking on the rideshare category. Again, the title, category, and date of post are displayed to the user. If the user would like to return back to the main page, they can click on the back button. If the user would like to view more details on a specific listing, they can click on the corresponding title to go to the detailed view.

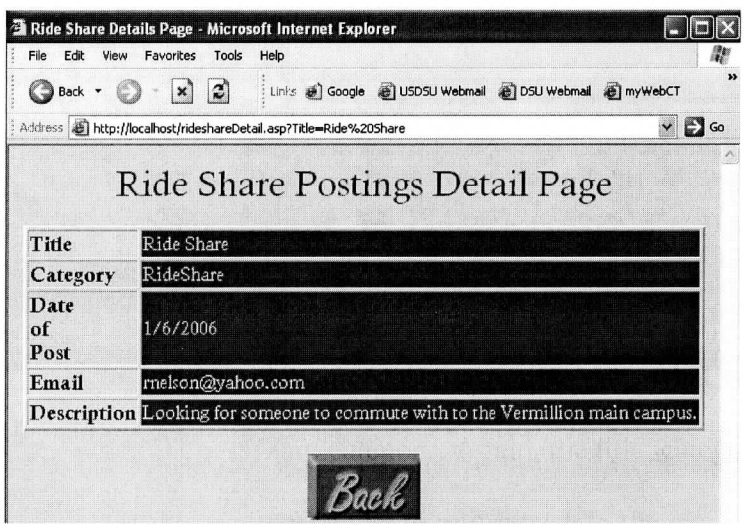


Figure 27: rideShareDetail.asp

Figure 27, rideShareDetail.asp, shows the detailed view for the user. The fields included in this view are title, category, name, email, and description. The user can easily locate the contact information in the detailed view in order to make the initial contact if interested in the item posted. There is also a back button for the user to easily navigate through the pages.

The following is the SQL code utilized to display only postings within the **textbook** category:

```
<%
Dim textbook
Dim textbook_numRows
Set textbook = Server.CreateObject("ADODB.Recordset")
textbook.ActiveConnection = MM_Bulletin_Board_STRING
textbook.Source = "SELECT Title, Category, Date_of_Post  FROM bulletin_posts
WHERE category LIKE 'textbooks'"
textbook.CursorType = 0
textbook.CursorLocation = 2
textbook.LockType = 1
textbook.Open()
textbook_numRows = 0
%>
```

The DIM statements in the first two lines of code are used to declare the variables "textbook" and "textbook _numRows. Set textbook = Server.CreateObject("ADODB.Recordset") is used to be able to read database data. Before the data can be read, it must first be loaded into a recordset which is done by the ADODB.Recordset statement in the code. The code textbook.ActiveConnection = MM_Bulletin_Board_STRING indicates to which connection object the specified recordset

object currently belongs. In this case, the connection is being made to the bulletin_board database.

The code includes SQL statements that are utilized to narrow the displayed results. The SELECT statement indicates which fields from a particular table to display. The fields that have been "selected" from the bulletin_post table are Title, Category, Name, Email, and Date_of_Post. The next statement in the query is the word FROM. This statement points to the table to pull the information "from." In this statement, the table the information is being taken from is the bulletin_posts table. The next portion of the statement is LIKE. The LIKE statement locates only records that match the particular field name; in this case we are displaying records from the "textbook" category. The final portion of the statement is ORDER BY. This particular statement determines how the records that meet the criteria are ordered. In this particular statement, the results are ordered by the Date_of_Post field.

The cursor type and lock type properties specified when opening a recordset is important. In the code textbook.CursorType = 0 the cursor type is set to the default zero. What this means when the code is executed is that this moves forward through the records. The reason the forward only method is utilized when moving through the records is for performance. This type of cursor moves more speedily through the records and takes less time to complete.

The lock type only allows data to be read from, but not altered. In this particular case, I want the user to be able to read, but not write to the records. In the code "textbook.LockType = 1", I am setting the locktype for this query to the value one which specifies a "read-only" attribute for this property. The cursor location property defines where the recordset is created when opened. In this line of code "textbook.CursorLocation = 2", the

cursor location is set to two which indicates that the recordset is created on the server-side cursor as opposed to the client-side cursor.

After this step of the code has completed, the next step is to OPEN the records and display them to the user. The Open method opens the database to allow access to records in a table, the results of a query, or to a saved Recordset. In this case, the only records that will be displayed are the ones that fall under the "textbook" category. The final portion of the code is "textbook _numRows = 0".

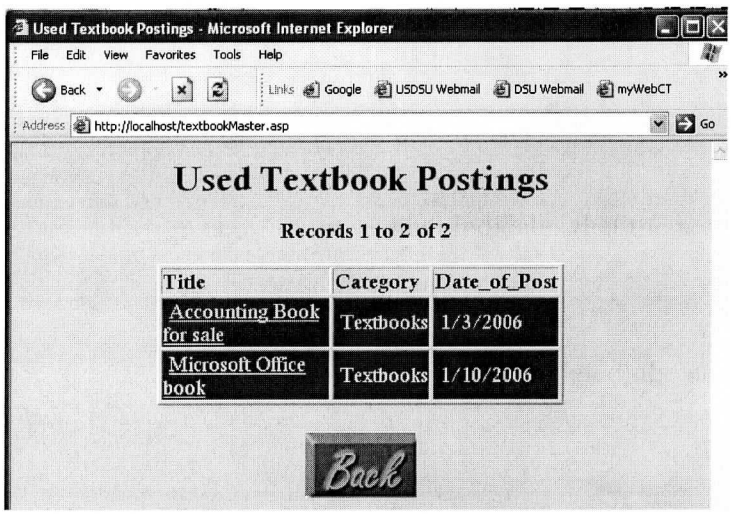


Figure 28: textbookMaster.asp

Figure 28, textbookMaster.asp, illustrates what is displayed to the user when mouse clicking on the textbook category. Again, the title, category, and date of post are displayed to the user. If the user would like to return back to the main page, they can click on the back button. If the user would like to view more details on a specific listing, they can click on the corresponding title to go to the detailed view.

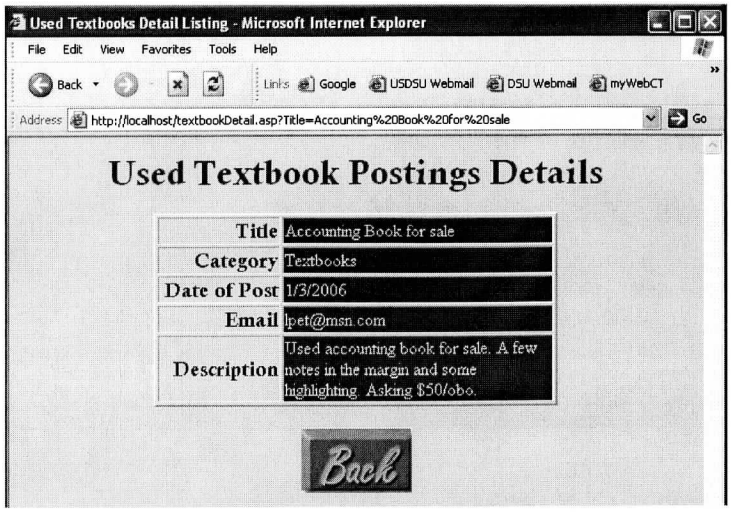


Figure 29: textbookDetail.asp

Figure 29, textbookDetail.asp, shows the detailed view for the user. The fields included in this view are title, category, name, email, and description. The user can easily locate the contact information in the detailed view in order to make the initial contact if interested in the item posted. There is also a back button for the user to easily navigate through the pages.

Figure 30 illustrates the "howtopost.asp" page included on the online bulletin board. If the user mouse clicks the "Navigating the Bulletin Board" link on the home page, the user will be provided instructions on how to view, post, and edit/delete.

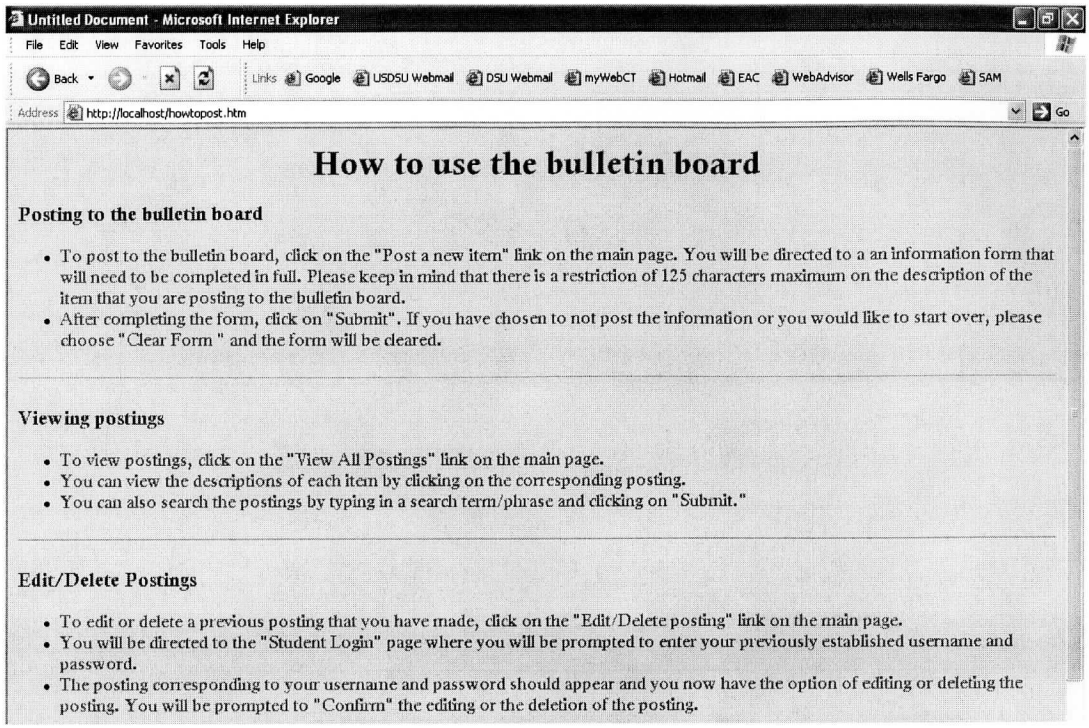


Figure 30: howtopost.asp

Another web-interface design I included was customized "Home" and "Back" hover buttons figures 31, 32, 33, and 34. Many pages within the bulletin board web site contain these particular buttons. I utilized Macromedia Fireworks to create my own hover buttons. I coded them to swap images with rollover effects. When the user moves their mouse over the buttons, the button's text turns yellow to indicate to the user that the button is active.

Figure 31: home button.png

Figure 32: home button2.png

Figure 33: back button.png

Figure 34: back button2.png

LITERATURE REVIEW

The literature review for this particular project was difficult. There is currently a lack of pertinent information regarding academic online bulletin boards such as the one described in this paper, especially for a non-traditional based campus such as USDSU.

The first located article discussed the shift of universities from selling books only through traditional methods to allowing students an opportunity to buy and sell used textbooks in an online forum. The article discussed a decision by the University of Miami to implement an online bulletin board to aid in reducing the amount of postings on the traditional bulletin boards. The University of Miami's bulletin board only allows students to post that have a valid university email. Universities, such as Georgetown and Harvard, have implemented similar online bulletin boards in response to growing requests by students for an online exchange forum (Johnson, 2005).

The second article located was published in a university newspaper for Binghamton University in Binghamton, NY. The article begins by discussing the inconvenience the traditional method of posting bulletin board announcements offers to students. The "e-Club" at the university created the "Bearcat Exchange" which allows students to browse many items for sale, rentals wanted, etc…which is similar to the bulletin board that I have created for USDSU. Binghamton University chose to create their online exchange forum originally in response to students in search of a better medium for textbook exchange and also to remove the middleman. In the long run, they believe that this site will save students valuable time and money while providing a more convenient and simplified way of conducting exchanges on their campus (Boyarsky, 2003).

Overall the literature that I located favored the online bulletin board instituted within the educational setting. It provides students with a centralized area for distribution of information. The one controversial area I encountered pertained to used textbooks. There was divergence within the university setting as to whether or not textbooks should be offered via bulletin boards and the negative affect for the bottom line for university bookstores.

CONCLUSIONS

When I began work on the INFS 788 project I really was naïve. I was not truly aware of how much time and effort this would require of me. I went into the project with high expectations for the outcome. I am satisfied with what I have achieved and have gained some invaluable experience through the implementation of this project.

This project entailed large amounts of coding. There were multiple pages included with the site to offer the users a multitude of methods of viewing the bulletin board information. There are minimal postings included at this time but this will change when the online bulletin board is implemented on the USDSU website. I took this into consideration when creating this website.

The main goals of the project were to:

1. Create a web presence that was easy to navigate for all users
2. Allow the user to browse the information in a customer-friendly manner,
3. Display the selected postings as requested, and
4. To have a website that loads quickly so that the user does not become agitated and becomes a frequent visitor to the online bulletin board.

There also was a large quantity of database controls that were required within the site. The first challenge was connecting the form in createPost.asp to the fields in the table bulletin_post. Another aspect pertaining to the database was proper display of information. Querying the table with Structured Query Language to display the correct information was an accomplishment. It was a terrific learning experience for me and I gained a new appreciation for the individual efforts put forth by programmers and database analysts.

It was difficult at times to foresee the end of the project and I feel for the most part that I have successfully completed what I set out to do. If I would have had more time and resources available, I certainly would have focused more on the visual design of the site. I would have liked to have developed the design a bit more fully. But my main focus from the beginning of this project was to create a form for the user to enter the data and have it post to a database as well as proper display of the bulletin board information. These tasks consumed most of my project implementation phase. It was rewarding when the web site and the bulletin board connections were finally in working order. Overall, I feel that I did achieve the main objectives that I set for myself at the onset of this project.

The anticipated deliverables were certainly achieved. This site is fully functional and after some tweaking can be implemented on the USDSU web site. This project will be beneficial to USDSU faculty, staff, and students. It has also served as an invaluable learning experience for me. This experience has allowed me to apply my skills and abilities to a real-world situation as well utilize the knowledge obtained from many courses completed in my graduate program including: INFS 720 Systems Analysis and Design Using Case-Based Tools, INFS 730 Programming for E Commerce, INFS 732 Electronic Commerce, INFS 752 Advanced Network Technology and Management, and INFS 760 Database Design and Performance Analysis.

REFERENCES

2004 – 2005 Accountability report. SD Board of Regents. Retrieved April 26, 2006, from http://www.sdbor.edu/publications/documents/AccountabilityReport.pdf.

Bakharia, K. (2002). Macromedia Dreamweaver MX: Fast & Easy Web Development Retrieved January 1, 2006, from www.netlibrary.com.

Boyarsky, A. (2003 May 2). Student online trading post opens for all sorts of swaps. *Pipe Dreams, vol. lxiv, issue 22*. Retrieved April 21, 2006, from http://www.bupipedream.com/050203/news/n3.htm.

Colorado College Handbook: Glossary (2001-2006). Colorado Council on High School/College Relations. Retrieved, April 29, 2006, from http://www.collegeincolorado.org/Expert/glossary.asp

FY2005 Report (2006). Sioux Falls: USDSU.

Gosselin, D. (2004). *Javascript.* (3rd edition). Course Technology.

Johnson, A. (2005, March 21). Students go online to swap, sell texts. *Washington Times*. Retrieved April 20, 2006, from http://www.washingtontimes.com/business/20050321-121134-1198r.htm.

MSDN Library. Retrieved April 26, 2006, from http://msdn1.microsoft.com/en-us/default.aspx.

Payne, C. (2002). *Sams Teach Yourself ASP.NET in 21 Days*. (2nd Edition). Sams.

W3Schools. Retrieved April 27, 2006, from http://www.w3schools.com/default.asp.

Wikipedia. Retrieved April 29, 2006, from www.wikipedia.org.

APPENDIX A

Work Breakdown Structure

1.0 Project Initiation

- 1.1 Meet with USDSU Staff to discuss project requirements
- 1.2 Complete Work Breakdown Structure (WBS)
- 1.3 Complete Gantt Chart

2.0 Project Planning

- 2.1 Present ideas to USDSU Staff
- 2.2 Review and adjust project to meet staff requirements
- 2.3 Meet with staff to finalize project plans
- 2.4 Begin drafting project plan
  - 2.4.1 Construct sketch of overall site
  - 2.4.2 Review and construct authentication schematics
  - 2.4.3 Review and establish database requirements
  - 2.4.4 Review coding requirements
  - 2.4.5 Review overall design requirements
  - 2.4.6 Write up rough draft of project plan
- 2.5 Present project plan to USDSU staff
- 2.6 Review staff comments and create second draft
- 2.7 Present second draft to USDSU Staff
- 2.8 Review staff comments and finalize project
- 2.9 Submit project plan to MSIS Program director

3.0 Project Execution

- 3.1 Create databases for authentication
- 3.2 Begin coding opening page
- 3.3 Present opening page to USDSU staff
- 3.4 Review comments and make adjustments
- 3.5 Begin coding layout for remaining pages
- 3.6 Present bulletin board to USDSU staff
- 3.7 Make any necessary adjustments and finalize

4.0 Project Closing

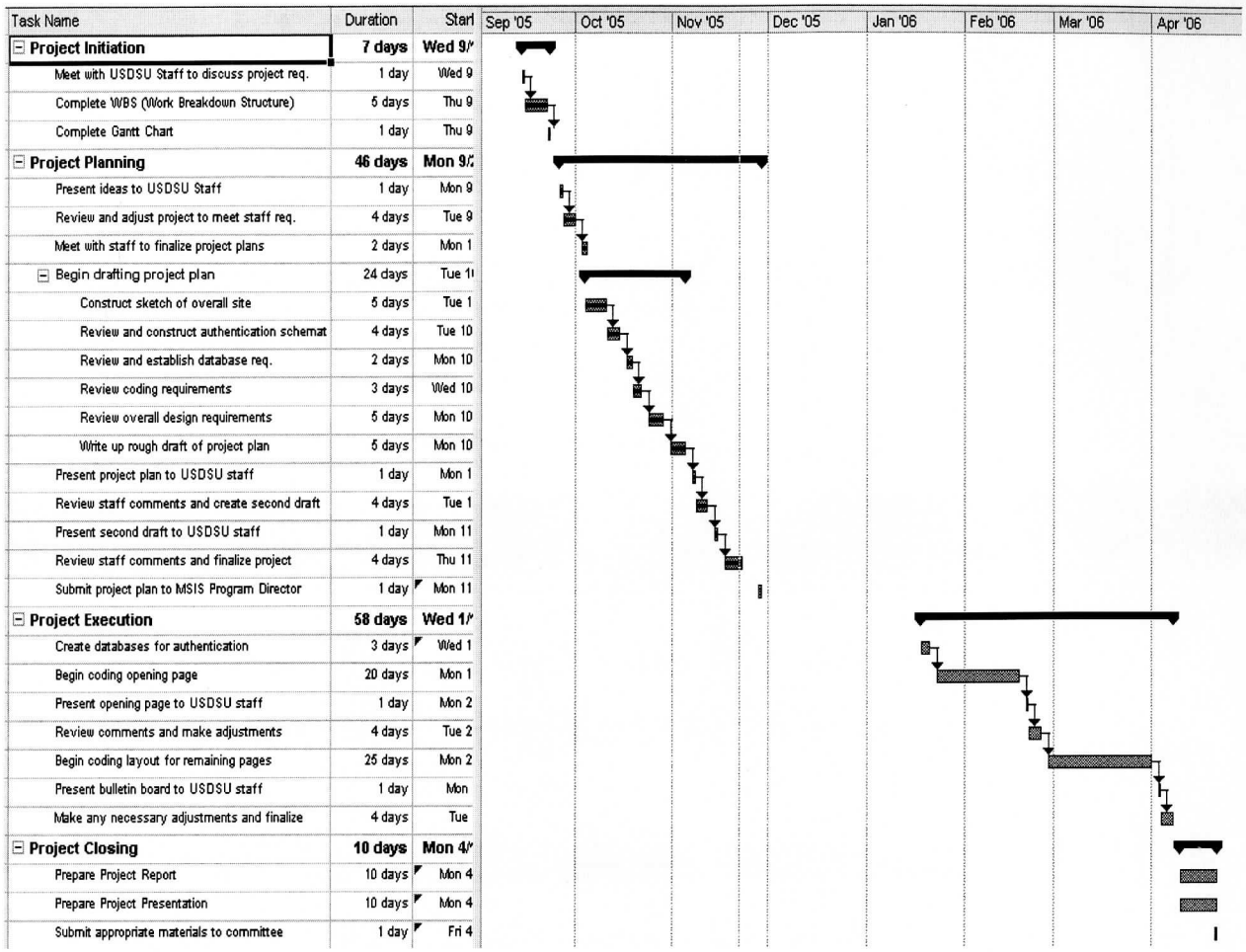
4.1 Prepare project report

4.2 Prepare project presentation

4.3 Submit appropriate materials to committee

APPENDIX B

Gantt Chart



Task Name	Duration	Start
Project Initiation	**7 days**	**Wed 9/**
Meet with USDSU Staff to discuss project req.	1 day	Wed 9
Complete WBS (Work Breakdown Structure)	5 days	Thu 9
Complete Gantt Chart	1 day	Thu 9
Project Planning	**46 days**	**Mon 9/**
Present ideas to USDSU Staff	1 day	Mon 9
Review and adjust project to meet staff req.	4 days	Tue 9
Meet with staff to finalize project plans	2 days	Mon 1
Begin drafting project plan	24 days	Tue 1
Construct sketch of overall site	5 days	Tue 1
Review and construct authentication schemat	4 days	Tue 10
Review and establish database req.	2 days	Mon 10
Review coding requirements	3 days	Wed 10
Review overall design requirements	5 days	Mon 10
Write up rough draft of project plan	5 days	Mon 10
Present project plan to USDSU staff	1 day	Mon 1
Review staff comments and create second draft	4 days	Tue 1
Present second draft to USDSU staff	1 day	Mon 11
Review staff comments and finalize project	4 days	Thu 11
Submit project plan to MSIS Program Director	1 day	Mon 11
Project Execution	**58 days**	**Wed 1/**
Create databases for authentication	3 days	Wed 1
Begin coding opening page	20 days	Mon 1
Present opening page to USDSU staff	1 day	Mon 2
Review comments and make adjustments	4 days	Tue 2
Begin coding layout for remaining pages	25 days	Mon 2
Present bulletin board to USDSU staff	1 day	Mon
Make any necessary adjustments and finalize	4 days	Tue
Project Closing	**10 days**	**Mon 4/**
Prepare Project Report	10 days	Mon 4
Prepare Project Presentation	10 days	Mon 4
Submit appropriate materials to committee	1 day	Fri 4